



Servicio con JSON

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso sobre Servlets. En este video vamos a implementar un web service. ¿Qué era un web service? La idea de un web service es que nosotros usando HTTP podamos ejecutar una función de nuestro servidor y que nuestro servidor nos devuelva un JSON o un XML.

[00:19] Entonces por ejemplo, si ven nosotros con nuestro Servlet podemos hacer eso, utilizando HTTP podemos ejecutar una función, solo que hoy en día nosotros estamos devolviendo un JSP, un HTML. Entonces, lo que nosotros tendríamos que hacer es tener alguna lógica ahí, en nuestro controlador, en el cual nos diga de que entonces estamos pidiendo un JSP, perdón, un JSON, entonces deberíamos devolver un JSON.

[00:49] Entonces tendríamos que tener, por ejemplo, en nuestra lista de empresas algo así como return "json", alguna cosa así y especificar cuál JSON nosotros tenemos que devolver, entonces es una lógica que no solamente vamos a tener que cambiar nuestras acciones, sino que también nuestro controlador también debería cambiar.

[01:10] Entonces en acá en vez de hacer un else, if, recibimos un JSON, entonces tenemos que hacer toda una lógica para obtener el JSON. No nos vamos a meter con eso que voy a dar un "Ctrl + Z" acá, no nos vamos a meter con eso. ¿Por qué? Porque ya frameworks de más alto nivel como Spring MVC, por ejemplo, ya hacen todo esa parte de forma mucho más fácil que nosotros haciéndolo manualmente.

[01:32] Entonces, vamos a pasar para el siguiente nivel directamente y vamos a crear un nuevo Servlet. Vamos a crear un nuevo Servlet solo para mostrar cómo realizar esa tarea de devolver un JSON. Entonces vamos a ir a nuestra carpeta de Servlet, clic derecho new Servlet. Vamos a colocar acá el nombre de `empresasService`.

[02:02] Va a tener una herencia de `HttpServlet`, next. Acá el URL Mapping va a ser `empresas` solamente, no va a ir para nuestra entrada, va a ser un servlet diferente. Siguiendo y acá va a tener un service. Saco el `doGet`, `doPost` y saco el constructor, que no lo voy a necesitar. Bien. Finish. Creamos acá nuestro Servlet, vean que no tiene la notación acá, entonces vamos a agregarla: `@WebServlet`, todo esto ya lo conocemos, ya lo sabemos.

[02:35] Va a estar viendo ("`/empresas`"), "`Ctrl + S`", este de acá lo saco y vamos a ir nuestro `web.xml`, que nos ha agregado acá nuestro servlet Mapping y lo vamos a eliminar. "`Ctrl + S`". Y ahora este servlet, la idea es que este `webService`, la idea es que devuelva un JSON o un XML. Vamos a comenzar con JSON.

[03:03] Entonces, para devolver un JSON primero necesitamos nuestras empresas, entonces, para esas empresas vamos a necesitar un `new DB()`, nuestra base de datos `.get`, tengo que importar, "`Ctrl + Shift + O`" importé la base de datos, `getEmpresas`, esto ya lo conocíamos. ¿Y ese `getEmpresas` devuelve qué? Una list de empresa, y vamos a llamarlo de empresas, es igual a "`Ctrl + Shift + O`" para importar. "`Ctrl + S`".

[03:35] Bien. Tenemos nuestras empresas. ¿Qué es lo que tendríamos que hacer ahora? Nuestra idea es convertir esas empresas al formato JSON. Y bueno, después devolverlo. ¿Ahora cómo es ese formato JSON? Para quienes nunca han visto ese formato, vamos a buscar acá `json example`, yo ya lo había buscado antes.

[03:59] Tenemos este JSON example acá, pero no me gusta, no lo veo tan claro, sobre todo por el fondo y todo eso. Una cosa interesante, JSON significa

JavaScript Object Notation, viene de la idea de JavaScript, un objeto en JavaScript se ve más o menos así como como aparece acá, pero voy a buscarles otro ejemplo que me gustó un poco más o queda un poco más claro. Voy a abrir esta imagen y vean lo siguiente.

[04:26] JSON comienza con llaves. Eso significa que es un objeto. Eso que está ahí dentro de una llave es un objeto. Así es como se define JavaScript. Y acá vamos a tener siempre llaves y valor, llave en el sentido no llave de esto, sino llave de key, en inglés key value, llave valor. En este caso por ejemplo, tendríamos una llave company y un valor mycompany. Entonces con la llave company yo puedo obtener el string mycompany.

[04:55] Acá, por ejemplo, un ejemplo de un objeto dentro del objeto JSON, entonces tendríamos “companycontacts” como llave, dos puntos, los dos puntos separan la llave del valor y tendríamos una llave. Una llave, que dentro de él, dentro de esa llave tenemos un objeto que está compuesto de dos atributos, dos llaves y dos valores. Bien.

[05:22] A su vez, por ejemplo, tenemos también un array acá para poder colocar una lista de cosas, en este caso lista de objetos, entonces tendríamos una lista de objetos, llaves, corchetes, corchetes para lista, llaves para objetos, entonces tenemos dos objetos, con coma vamos a separar esos objetos. Y bueno es todo eso.

[05:45] No hay mucha diferencia, podemos tener diferentes valores, tipos de datos, podemos tener int, stream, null. Entonces es simplemente eso. Es bastante, bastante simple, tal vez al comienzo pueda impactar, pero es simple créanme. ¿Ahora, cómo hacemos para convertir nuestro objeto, nuestra lista de empresas a JSON? Porque parece ser un una tarea, una tarea un poco difícil, digamos.

[06:15] Sería tener que agarrar cada objeto, obtener cada atributo, convertirlo en un string. Sería muy, muy molesto. Me gustaría tener una biblioteca que

pueda hacer eso automáticamente. Y por suerte existe una biblioteca. En realidad existen varias bibliotecas, muchas, pero en este caso yo voy a buscar una biblioteca se llama Gson.

[06:36] Gson viene de Google, es una biblioteca creada por Google. Y entonces, si nosotros vamos a la documentación oficial acá, nosotros vamos a poder ver, por ejemplo, cómo utilizar esta biblioteca. Si vamos dentro de github.com/google/gson, nos podemos ir a user guide y acá en object examples tenemos un objeto de cómo utilizar esa biblioteca Gson.

[07:03] Entonces esta parte de acá es la que a mí me importa que es la conversión realmente, en realidad la creación del objeto GSON y después la conversión usando un método de ese objeto GSON, y nosotros enviamos un objeto, nuestro objeto, en este caso sería la lista de empresas.

[07:20] Entonces vean que es muy, muy fácil. Y vamos a hacer eso. Ahora copié este código que tenía cada ejemplo, pegué GSON y acá en vez de objeto vamos a tener nuestras empresas. GSON ya sabe cómo interpretar objetos Java y convertirlos automáticamente a JSON, una maravilla, ahora no tenemos GSON.

[07:47] GSON no es una biblioteca que viene por patrón, así por default en nuestra aplicación web. Entonces nosotros vamos a tener que importar una esa biblioteca ¿cómo lo hacemos? A través de nuestras libs, nosotros vamos a poner esa biblioteca, yo ya tengo acá esa biblioteca bajada, GSON.

[08:09] Entonces voy a copiar esa biblioteca, "Ctrl + C", bien, y me voy a mi lib, "Ctrl + V". Con esto, nosotros ya tendríamos GSON y acá, si yo me voy a los problemas que está dando, dice import GSON. Con esto ya me agregé acá el import y ya estaría esta parte funcionando por lo menos hemos convertido nuestras empresas a JSON.

[08:40] ¿Ahora, qué es lo que tenemos que hacer? Nosotros tenemos que devolver ese objeto a quien lo pidió. ¿Cómo hacemos eso? Nosotros ya lo hemos hecho anteriormente. Vamos a usar `response.getWriter`, se acuerda al

comienzo del primer curso, usamos ese `getWriter` para devolver strings.

Entonces en este caso, ¿nosotros vamos a devolver qué cosa? Vamos a devolver este JSON.

[09:04] Bien, bien. Vamos a usar `response.getWriter().print` y acá adentro nosotros vamos a colocar nuestro (json); este método `getWriter.print` ya lo habíamos usado en el primer curso de servlets, para devolver unos strings. Entonces acá usamos eso mismo, pero devolvemos nuestro Jason, ya que Jason es simplemente un string muy largo.

[09:36] ¿Qué es lo que ocurre? Nosotros podríamos devolver esto así, pero nosotros también tenemos que decirle a nuestro cliente cuál es el tipo de formato, cuál es el tipo de datos que nosotros estamos devolviendo. Porque, por ejemplo un Server, un servidor podría devolver un PDF, HTML, podría devolver solamente un string, JSON por ejemplo.

[10:00] Nosotros tenemos que avisarle de una forma cordial que: “mira. Vos vas a estar recibiendo un objeto del tipo JSON, los datos que te estoy devolviendo son del tipo JSON”. Eso para que pueda saber interpretar cómo es que tiene que interpretar esa respuesta. Bien, ¿entonces cómo hacemos eso?

[10:21] Nosotros. Perdón, nosotros no. Existe un grupo que se encarga de ver cuáles son todas estas normas y formas de comunicarse dentro del Protocolo HTTP y ellos dijeron que tienen que ser de la siguiente forma: `response.setContentType` y acá en el tipo, esto va a ser un encabezado que va a estar dentro del Protocolo HTTP.

[10:54] Entonces es `ContentType` tiene que ser del tipo (“`Application/json`”). Con esto nosotros en ese encabezado le estamos diciendo que nuestra respuesta que estamos enviando es del tipo JSON. Si nosotros quisiéramos enviar una respuesta del tipo XML, sería XML. Con HTML, por ejemplo sería (“`Text/html`”), pero por ejemplo, en nuestro JSP no necesitamos especificar que

vamos a devolver un HTML porque ya nuestro Tomcat, nuestro JSP lo hace automáticamente.

[11:32] Pero entonces, volviendo, volví muy para atrás. Volviendo, acá voy a colocar ("Application/json") "Ctrl + S" y vamos a ver qué es lo que ocurre. Voy a reiniciar el servidor porque hemos importado una biblioteca, hemos hecho varias cosas. Entonces bien, vean la consola, si no hay nada raro, parece que no.

[11:59] Vamos a intentar ver cómo funciona esto. Localhost, gerenciador y acá en vez de entrada vamos a ir a empresas, vamos a ver si funciona. Okay, vean que funcionó. Veán lo siguiente. Nos ha devuelto un JSON largo, porque es un string largo en el cual tenemos una lista de objetos, también tiene un corchete, inicia y finaliza con un corchete, y tiene un objeto, nuestra primera empresa y nuestra segunda empresa que es Alura y Caelum.

[12:28] Veán cómo se ha hecho automáticamente la conversión de los datos. Tenemos un id, un nombre y nuestra fecha de abertura, que esa fecha de abertura podríamos usar un formador para mejorar la fecha, etcétera, pero la base está, hemos podido devolver todo. Ahora, por ejemplo, me gustaría hacer lo siguiente, me gustaría mostrarles de una forma más bonita.

[12:50] Si ustedes hacen F12, abren la consola, como ya lo hemos visto anteriormente, vamos a dar un F5 y vean, dentro de nuestras empresas, acá se ve de una forma más bonita ese JSON. Google Chrome y todos los navegadores en sí ya traen esa forma de formatear los datos del tipo JSON.

[13:14] Y no solo eso, sino que vean lo siguiente, nosotros en nuestro header, acá vemos que el response, el tipo de datos es application/json y nuestro navegador ha pedido, por ejemplo los tipos text/html, application/html, etcétera. Ha pedido varios tipos diferentes.

[13:38] Perfecto. Entonces, con esto hemos terminado esta parte de JSON y vamos a ver en el próximo video cómo hacer para devolver un XML. Nos vemos

allá.